MINTS: Unsupervised Temporal Specifications Miner

Pradeep Kumar Mahato, Apurva Narayan
Department of Computer Science
The University of British Columbia
BC, Canada

Outline

- Introduction
- Objective
- Methodology
- Experimental result
- Conclusion and future work

Introduction

“Software specifications define system behavior, usage guidelines, program requirements and act as a tool for debugging..”

```
package androidx.appcompat.app;

import ...

Base class for activities that wish to use some of the newer platform features on older Android devices. Some
of these backported features include:
• Using the action bar, including action items, navigation modes and more with the
  setSupportActionBar(Toolbar) API.
• Built-in switching between light and dark themes by using the Theme.AppCompat.DayNight theme and
  AppCompatDelegate.setDefaultNightMode(int) API.
• Integration with DrawerLayout by using the getDrawerToggleDelegate() API.
Note that every activity that extends this class has to be themed with Theme.AppCompat or a theme that
extends that theme.
Developer Guides
For information about how to use the action bar, including how to add action items, navigation modes and
more, read the Action Bar API guide.
public class AppCompatActivity extends FragmentActivity implements AppCompatCallback,
        TaskStackBuilder.SupportParentable, ActionBarDrawerToggle.DelegateProvider {

    private AppCompatDelegate mDelegate;
    private Resources mResources;

    Default constructor for AppCompatActivity. All Activities must have a default constructor for API 27 and
    lower devices or when using the default android.app.AppComponentFactory.
    public AppCompatActivity() { super(); }
```

Source: Photo by Rami Al-zayat, Kelly Sikkema, Markus Spiske and Sebastian Herrmann on Unsplash

Objective

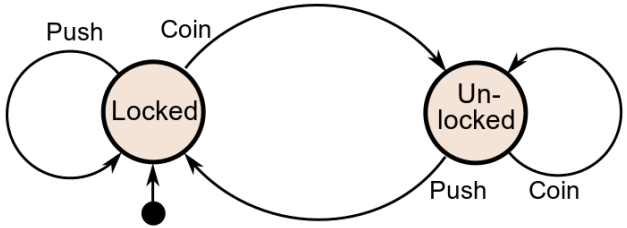
Mine system behavior from execution logs.

A popular approach for mining system behavior uses a finite state machine. Finite State Machines represents the working of a system and its modules using graphs. State is represented as nodes and edges are used to define change in state on certain action.

Figure: A turnstile

Behavioural Representation

Figure: Finite State Machine of a turnstile



Source: By Sebasgui - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=8930080

Source: By Chetvorno - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=20269475

Preliminaries

System activities are events. Traces are a collection of events.
An alphabet is a finite set of events.

Alphabet (Σ) → { $event_1$, $event_2$, ... }

Trace

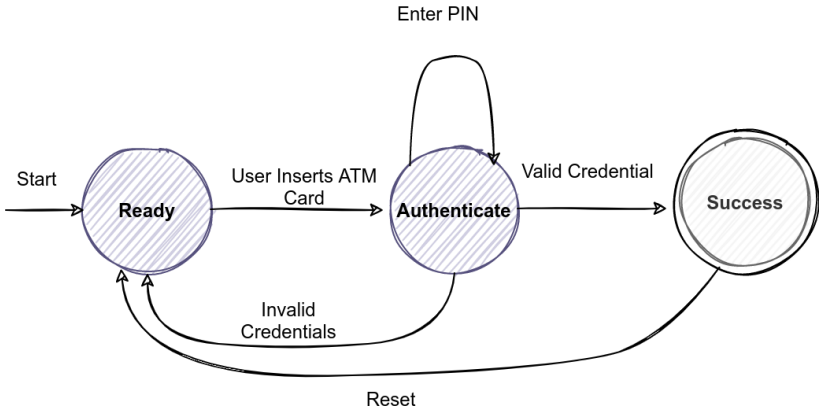
Group of events forms a trace

```
Sep  9 12:24:27 pradeep-Dell-System-XPS-L502X pkexec[21417]: pradeep: Executing command [USER=root] [TTY=unknown] [CWD=/home/pradeep] [COMMAND=/usr/lib/gnome-settings-daemon/gsd-backlight-helper --set-brightness 15]
Sep  9 12:24:27 pradeep-Dell-System-XPS-L502X pkexec: pam_unix(polkit-1:session): session opened for user root by (uid=1000)
Sep  9 12:24:27 pradeep-Dell-System-XPS-L502X pkexec[21422]: pradeep: Executing command [USER=root] [TTY=unknown] [CWD=/home/pradeep] [COMMAND=/usr/lib/gnome-settings-daemon/gsd-backlight-helper --set-brightness 15]
Sep  9 12:24:27 pradeep-Dell-System-XPS-L502X pkexec: pam_unix(polkit-1:session): session opened for user root by (uid=1000)
Sep  9 12:24:27 pradeep-Dell-System-XPS-L502X pkexec[21427]: pradeep: Executing command [USER=root] [TTY=unknown] [CWD=/home/pradeep] [COMMAND=/usr/lib/gnome-settings-daemon/gsd-backlight-helper --set-brightness 15]
Sep  9 12:26:57 pradeep-Dell-System-XPS-L502X gnome-keyring-daemon[2509]: asked to register item /org/freedesktop/secrets/collection/login/113, but it's already registered
Sep  9 12:54:18 pradeep-Dell-System-XPS-L502X pkexec: pam_unix(polkit-1:session): session opened for user root by (uid=1000)
Sep  9 12:54:18 pradeep-Dell-System-XPS-L502X pkexec[24982]: pradeep: Executing command [USER=root] [TTY=unknown] [CWD=/home/pradeep] [COMMAND=/usr/lib/gnome-settings-daemon/gsd-backlight-helper --set-brightness 5]
Sep  9 13:17:01 pradeep-Dell-System-XPS-L502X CRON[27846]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep  9 13:17:01 pradeep-Dell-System-XPS-L502X CRON[27846]: pam_unix(cron:session): session closed for user root
```

Figure: Snapshot of authentication log from Ubuntu 18.04

Timed Automaton

A finite state machine with **time sensitive** transition.



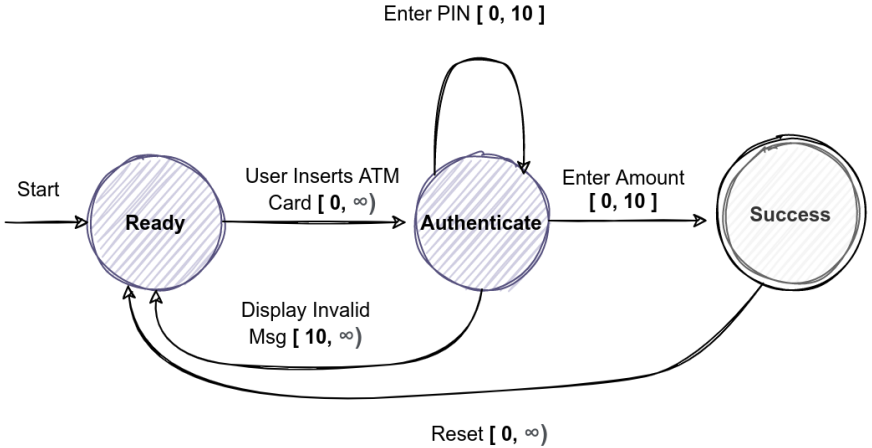


Figure: Finite State Machine of an ATM (Automatic Teller Machine)

Figure: A *timed* Finite State Machine of an ATM (Automatic Teller Machine)

Methodology

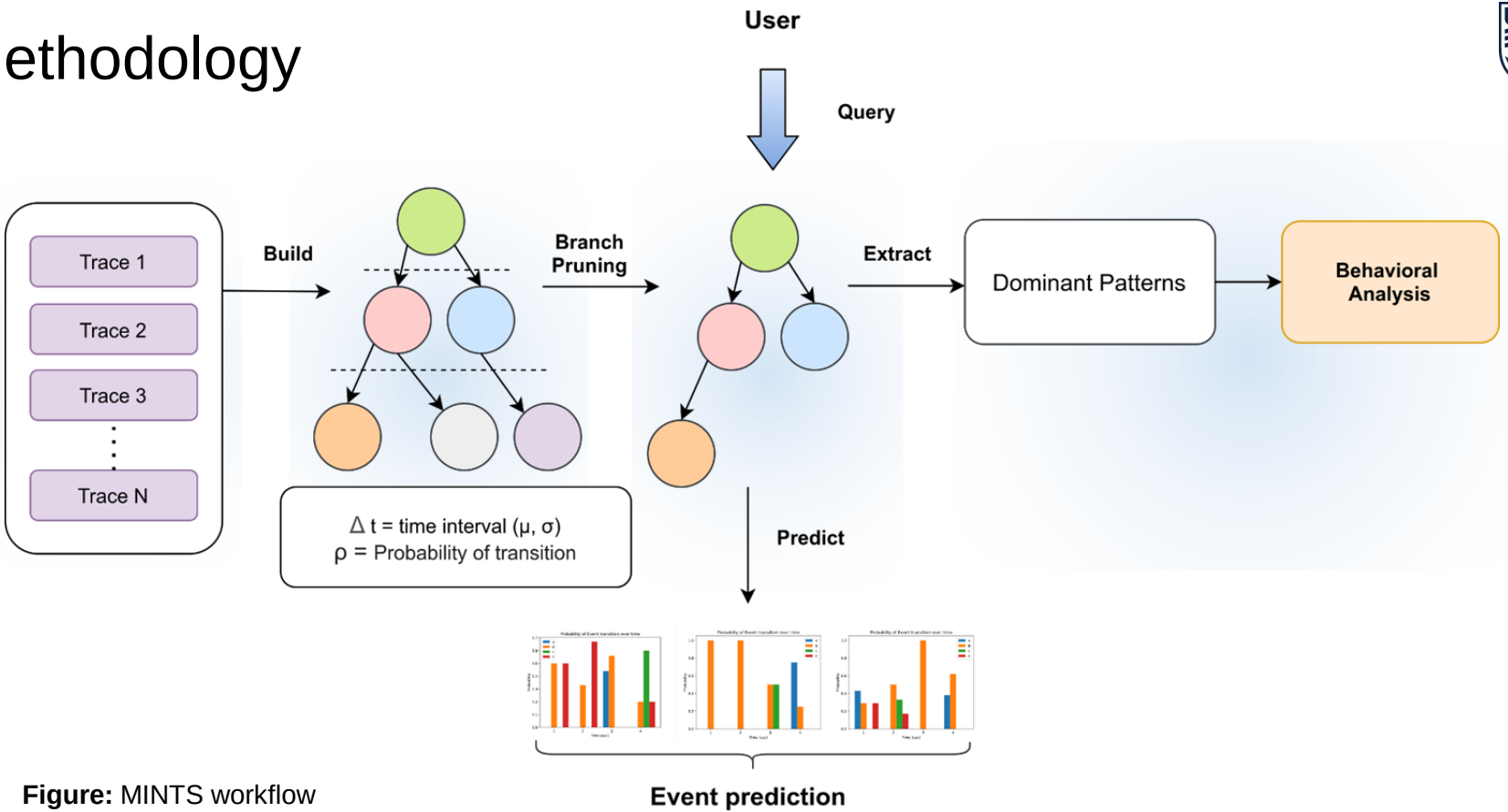
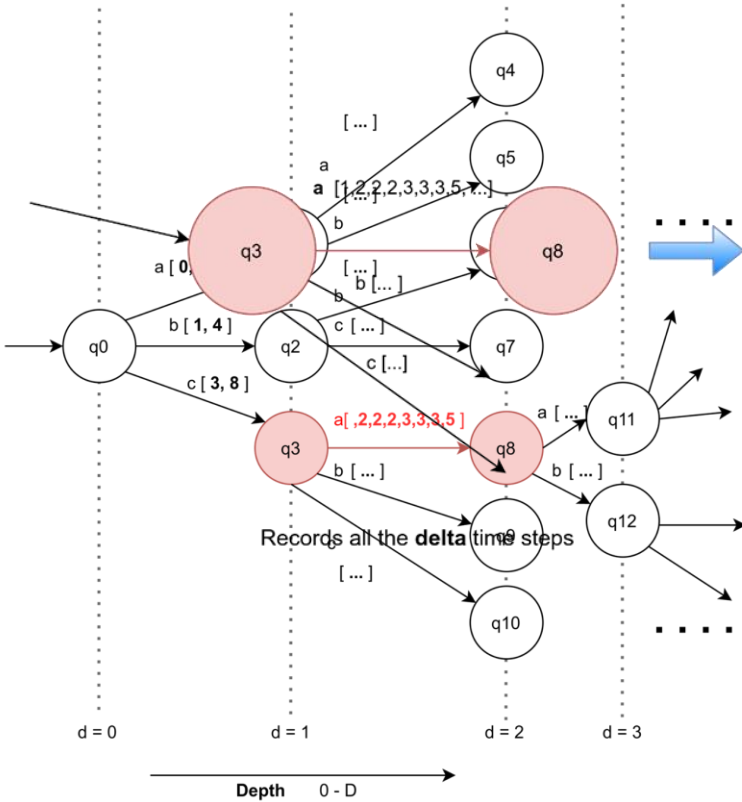
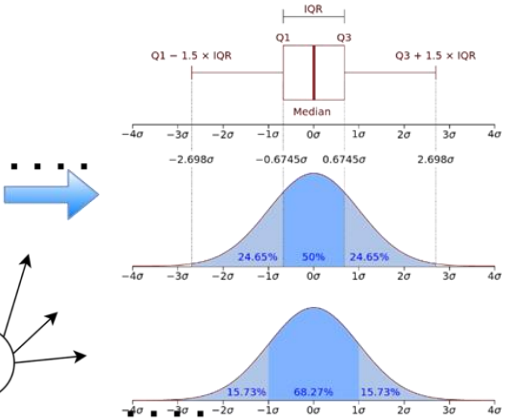


Figure: MINTS workflow

Determining Clock Constraint

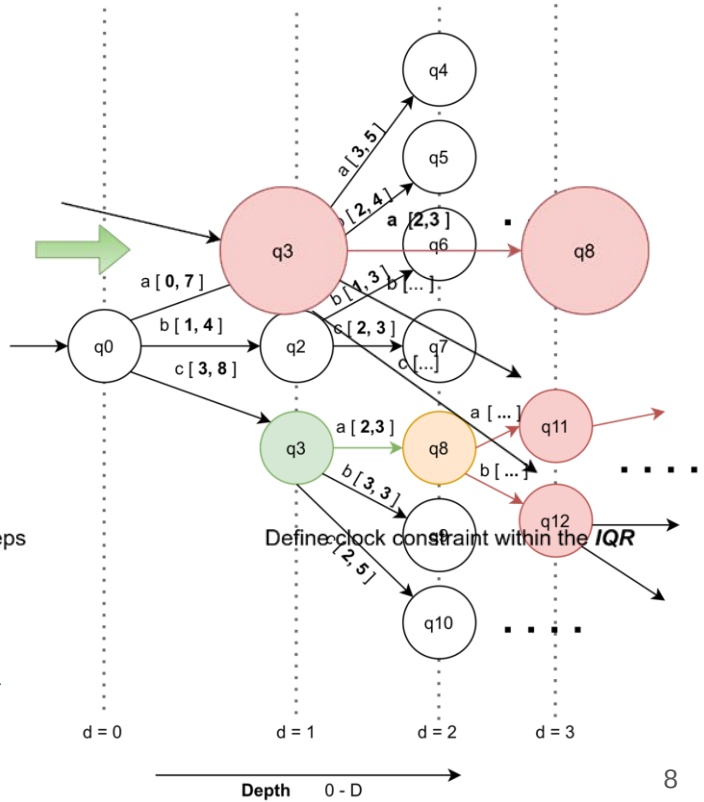


Records all the **delta** time steps



Normal Distribution over the **delta** time steps

Clock constraint evaluated at d = 2

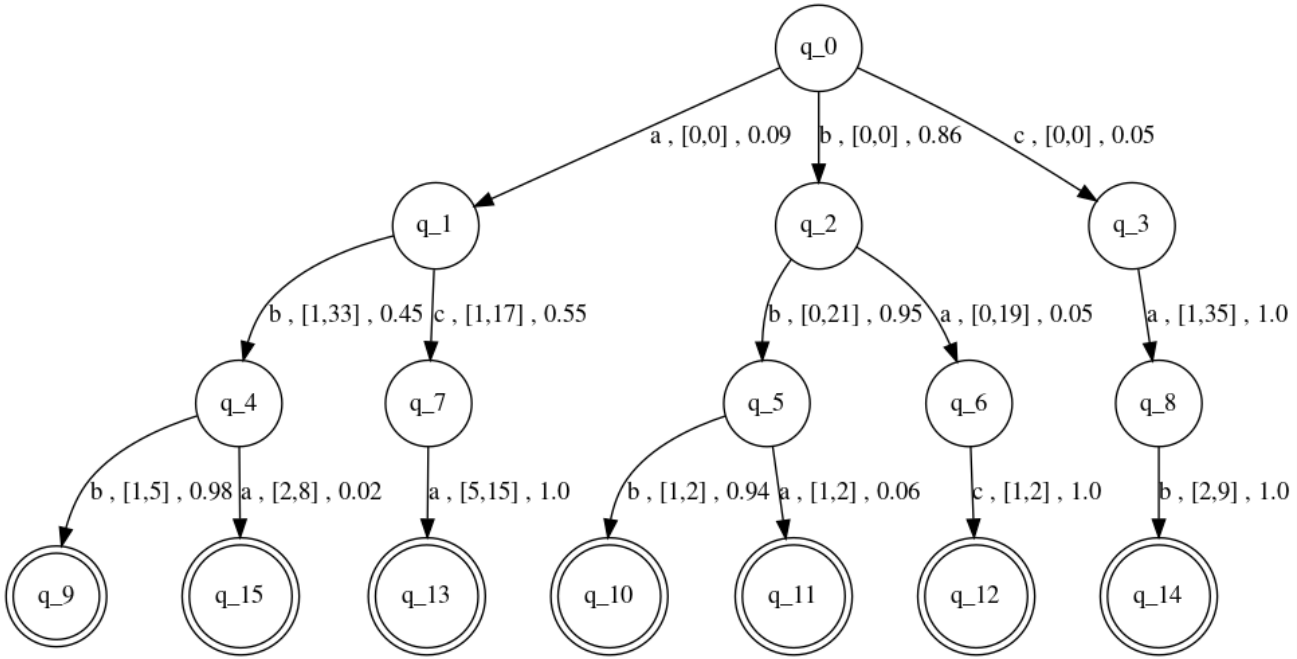


Define clock constraint within the ***IQR***

Extracted Timed Automaton

Nodes represent a state and edges contains the condition for transition.

Each edge, contains the triggering event, acceptable clock range and probability of transition.



Please Note: The probability of transition is different than event prediction (and the probability of event with time).

Experiments and Result

Objective

- Extract a timed automaton of the system
- Extract dominant temporal patterns

Measurement Metrics **(Association Rule Mining)**

- Support: Probability of a certain pattern in the data

  Example: 70% of customers buy milk and cereal

- Confidence: Probability of a certain event, given a series of events has already occured

  Example: 100% of customers who buys cereal also buys milk

Source: Photo by Visual Stories || Micheile on Unsplash

Dataset

- QNX Operating System Trace collected from a Hexacopter Drone
- Various flight maneuvers were performed
- 600+ Thousand events

Figure: Hexacopter Drone

Source: Photo by android99 on Free3d

Figure: Drone controller

Source: Photo by Ian Baldwin on Unsplash

Extraction of Dominant Temporal Patterns

Serial	Pattern	Count	Support	Confidence
1	[-INF,0]SND_MESSAGE [1373,9332]THREPLY	16234	0.029	0.988
2	[-INF,0]SYNC_CONDVAR_WAIT_82 [3222,156765]MSG_SENDV_11 [0,0]SND_MESSAGE [576,5867]THREPLY	2042	0.003	0.999
3	[-INF,0]THWAITPAGE [647,1773]THRUNNING [677,6843]REC_PULSE	1744	0.003	0.914
4	[-INF,0]MSG_READV_16 [26259,85259]MSG_REPLYV_15	623	0.007	0.257
5	[-INF,0]MSG_WRITEV_17 [2087,15713]MSG_WRITEV_17 [3487,20070]SND_PULSE_EXE	113	0.000	1.000

Table: Top 5 Dominant Temporal Properties extracted from the system trace

Hyperparams: $\mathcal{P}$ = 0.45, $\mathcal{D}$ = 10
Total Patterns Mined: 384

Extraction of Temporal Behavior

Specific temporal behavioral patterns were extracted that showed relationship causal relationship among the events.

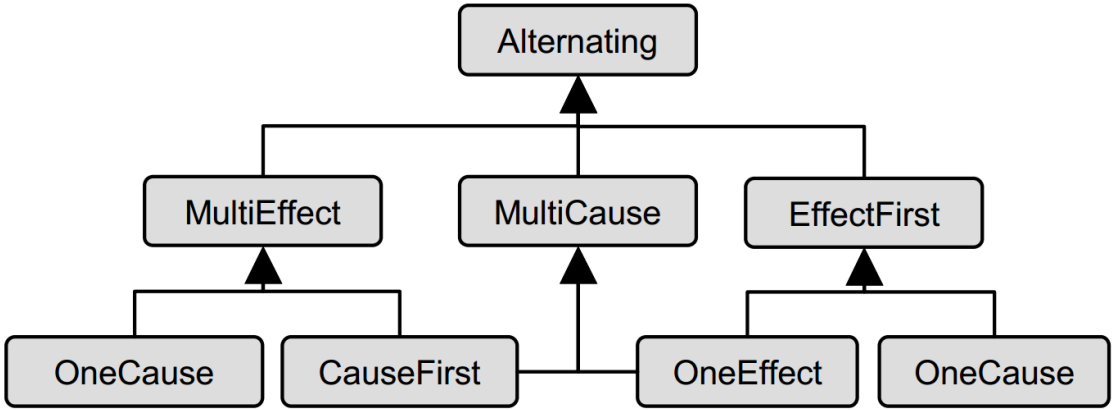


Figure: Temporal Behavioral Patterns [Yang et al, 2004]

Extraction of Temporal Behavior

Type	From-To Event	Pattern	Count	Support	Confidence
Response Pattern	SND_MESSAGE - THREPLY	[-INF,0]SND_MESSAGE [1373,9332]THREPLY	16234	0.029	0.988
	THREADY - MSG_REPLYV_15	[-INF,0]THREADY [371,1246]MSG_REPLYV_15	11069	0.029	0.367
	EVENT_2 - MSG_RECEIVEV_14	[-INF,0]EVENT_0 [0,0]MSG_RECEIVEV_14	1496	0.002	0.499
Alternating Pattern	THRECEIVE - THRUNNING	[-INF,0]THRECEIVE [306,640]THRUNNING	9040	0.068	0.996
	SND_PULSE_EXE - REC_PULSE	[-INF,0]SND_PULSE_EXE [2109,9901]REC_PULSE	6886	0.015	0.110
	SYNC_CONDVAR_WAIT_82 - MSG_SENDV_11	[-INF,0]SYNC_CONDVAR_WAIT_82 [3222,156765]MSG_SENDV_11	2061	0.003	0.453
Multi Cause	TIMER_TIMEOUT_75 - EVENT_1	[-INF,0]TIMER_TIMEOUT_75 [1330,22081]EVENT_1	1490	0.002	0.452
	MSG_READV_16 - MSG_REPLYV_15	[-INF,0]MSG_READV_16 [26259,85259]MSG_REPLYV_15	623	0.007	0.257
	CONNECT_FLAGS_43 - MSG_SENDV_11	[-INF,0]CONNECT_FLAGS_43 [824,3063]MSG_SENDV_11	36	0.000	0.494
Multi Effect	TIMER_TIMEOUT_75 - EVENT_1	[-INF,0]TIMER_TIMEOUT_75 [1330,22081]EVENT_1	1490	0.002	0.452
	REC_PULSE - BUFFER	[-INF,0]REC_PULSE [925,2049]BUFFER	28	0.001	0.013
	THWAITPAGE - THRUNNING	[-INF,0]THWAITPAGE [647,1773]THRUNNING	768	0.003	1.000

Table: 4 Temporal Behavioral Properties were extracted and validated

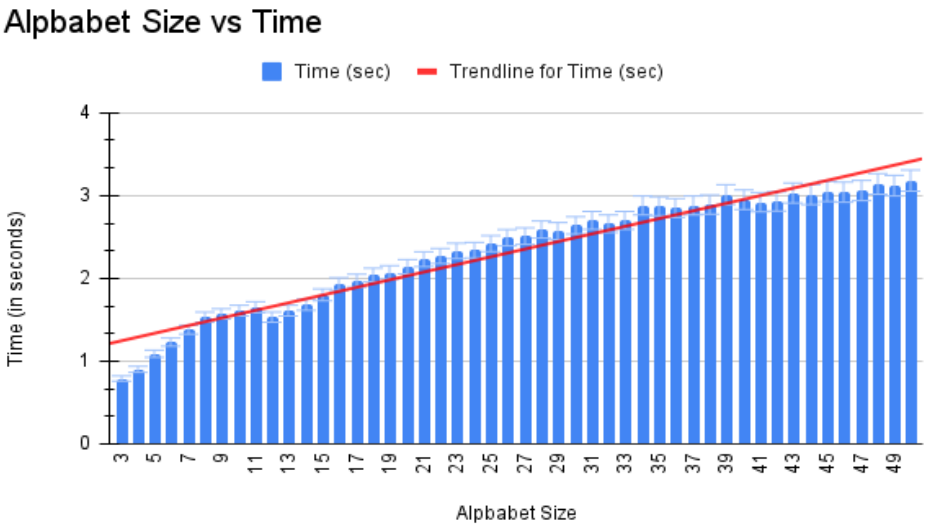
Feature Comparison

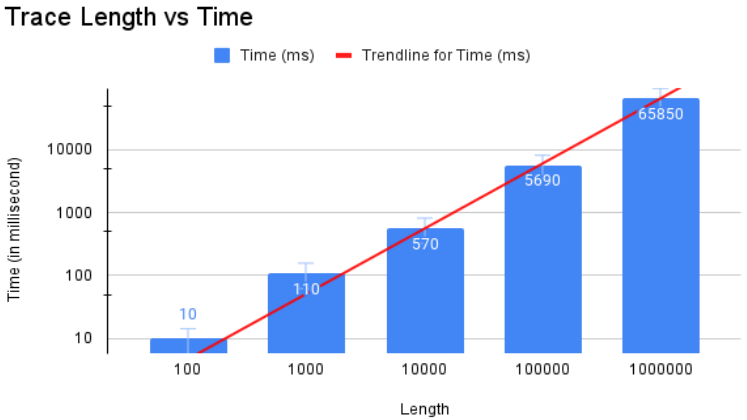
Algorithm (Author names used if algorithm has no name)	**Features**						
	Pattern Extraction	**Timed Pattern Extraction**	**Pattern Monitoring**	**Timed Pattern Monitoring**	**DFA Extraction**	**Timed DFA Extraction**	**Event Prediction**
UPPAAL SMC			✔	✔			✔
Texada	✔		✔				
Weiss et al.			✔	✔	✔		
J. An et al.					✔	✔	
MONTRE			✔	✔			
Cutulenco et al.	✔	✔	✔	✔			
QMine	✔		✔				
MINTS	✔	✔	✔	✔	✔	✔	✔

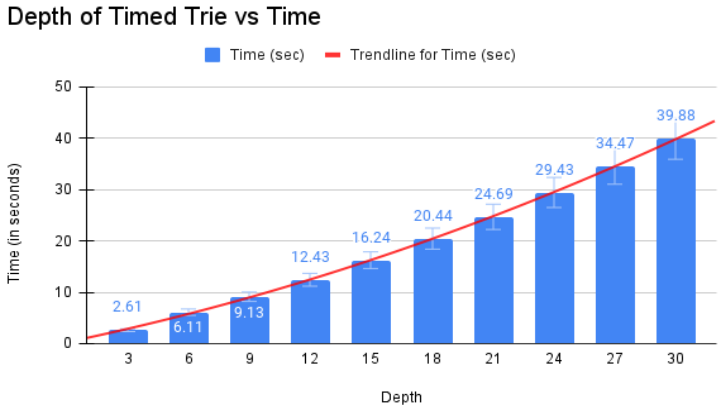
Table: Feature comparison with other prominent specification mining framework

Performance

Total time to mine: $\mathcal{O}(nD - D^2 + \Sigma^D)$







Conclusion and future work

- MINTS is capable of mining as well as monitoring of both timed and untimed specification
- Unsupervised learning of system behavior
- MINTS presents the learned model as Graphs
- MINTS is highly scalable
- MINTS has 100% theoretical soundness and completeness

Future Work

- Represent simpler timed-automaton using MINTS
- Simplify usage of the framework

Source: Photo by Simon Abrams on Unsplash

repo: https://github.com/Idsl-group/MINTS

References

- P. K. Mahato and A. Narayan. Qmine: A framework for mining quantitative regular expressions from system traces., QRS, 2020
- Dana Angluin. Learning regular sets from queries and counterexamples. Volume 75, page 87{106, USA, November 1987. Academic Press, Inc.
- A. Narayan, N. Benann, and S. Fischmeister. Mining specifications using nested words. In 2017 6th International Workshop on Software Mining (SoftwareMining),
- Apurva Narayan, Greta Cutulenco, Yogi Joshi, and Sebastian Fischmeister. Mining timed regular specifications from system traces. volume 17. Association for Computing Machinery
- Dogan Ulus. Montre: A tool for monitoring timed regular expressions. In Rupak Majumdar and Viktor Kunˇcak, editors, Computer Aided Verification
- Konstantinos Mamouras, Mukund Raghothaman, Rajeev Alur, Zachary G. Ives, and Sanjeev Khanna. Streamqre: modular specification and efficient evaluation of quantitative queries over streaming data

Thank You

Intentionally left blank